DWeb Weekend 2025 - Saturday 8/16/2025

Join this Signal group if you're interested in this topic: https://signal.group/#CjQKILX2PRzT1zkkX1MxegpnEveKtCbIgVMgiIMzSLYkgc06EhD6Z8HDUhc5hLWxK14ml910 (https://signal.group/#CjQKILX2PRzT1zkkX1MxegpnEveKtCbIgVMgiIMzSLYkgc06EhD6Z8HDUhc5hLWxK14ml910)

Roundtable Discussion 2.2

Title: *Data structures for P2P applications*
Time: *14:00 - 14:45*
Attendees: Bengo, Bhaskar, Suma, Steven, Liam / Steve, adz, Santi, Matthew, Dmitri, Matt, Duke
Note taker: adz

Please add your /initials/ at the front of a comment

General Notes

- Our discussion in the seminar was around data-centric networking
    - Don't think about the connectivity substrate, focus on the data structure instead
- Ensure integrity, provenance (Signature)
- Efficiency ("waiting for peers")
- We ask around why people are interested in this
    - Matthew
        - Different layers to this data stack
        - Thinking about the interface between "what the network is giving me" and the "actual materialized view"
        - Translating the operations I get from the network into the google

document ⇒ What is the interface between

    - adz
        - Support Internet and "Post-Internet" at the same time!
    - Liam
        - Interested in understanding the problem more
        - not familiar with CRDTs, I'm a cryptographer
    - Matt
        - Same, trying to understand the abstractions peoples are thinking in
    - Dmintri
        - Local first storage, requiring sync
        - You need to be aware of data structures
    - Bhaskar
        - First time, learning about p2p, efficient algorithms
    - Duke
        - Working on user-owned data methodologies, identity, security
    - Steven
        - Interested in what came out of the seminar
    - Suma
        - Interested in finding p2p operations which are easy & practical
    - Bengo
        - Decentralised social networking stuff, like Activity Pub, prev. Protocol Labs

- Who's unfamiliar with CRDTs? Do we need an introduction?
- We've identified same properties across projects (Holochain, p2panda, Hyper Hyper Space, TinySSB, ..)

    - We want to share this convergence

- "To make a sandwich you must first create the universe"
- Santi gives an intro
    - Concurrency, messages come in "out of order"
    - Logs help with this issue, we hash every entry and point at this hash in the next one
        - How do call this? "append-only log" or "forest of logs"
        - linked list with backpointers at the previous entry
            - Every dot in the graph is an "operation", the arrow is the pointer at the previous one
        - This gives us a sense of ordering, total order
    - DAGs
        - Directed Acyclic Graph
        - Same, but giving us a sense of concurrency, partial order, "comes before" relationship, causal order
        - Can't trust the wallclock time of your machine, we need to rely on this causal ordering
    - Single-Writer vs. Multi-Writer
    - Write everything into one log: SSBv1 vs. Write into different logs: SSBv2
        - Nice for avoiding "head of line blocking", prioritize some logs over others
    - It's coordination free!
        - CAP theorem: always available, this is why we call it "local-first"
    - Every author will have their own state where they data is
    - Make operations commutative when operations are concurrent
        - This is where the CRDT comes in!
    - If not always possible ..?

        - Not everything can be solved with an CRDT ..
        - They also tend to "grow" forever (unbound)
    - Let's look at an "grow only set" CRDT as an example
- When people said CRDTs usually meant "monotonic state"
    - But there's many cases where it is actually not, it's usually more a "coordination free" thing where you need to react to detected "conflicts" in non-monotonic ways
- There can be adversaries, trying to break the state of the network
    - Not perfectly solvable, depends on the trust model
- Automerge, Loro, Yjs are good CRDT libraries
    - But they are monolithic stacks
    - A lot of overlap in similar assumptions
    - If we start to agree on the properties we need from the underlying "data layers" we could imagine interoperatbility and "stacking"
        - For example "pull in the key agreement / group encryption library, some pruning technique and collaborative text"
- Bengo: RFC 6920!
    - Core DAG should be parameterized for Hashing & Signing
    - People disagreeing on these primitives hinders interop
    - Let's use the IETF one and not IPFS standards

Summary to be reported to the other DWeb Weekend participants

a) We've did an intro into CRDTs
c) Can there be core properties we can agree on to allow interop / layering?
b) Insights how it went for the IETF and IPFS standardization efforts trying something similar

adz